FAKULTÄT FÜR WIRTSCHAFTSINFORMATIK
UND WIRTSCHAFTSMATHEMATIK

UNIVERSITÄT MANNHEIM

Prof. Dr. Leif Döring
Daniel Schmidt, Benedikt Wille

Reinforcement Learning

8. Exercise Sheet

27.04.2026

For this assignment, knowledge from lectures 1 to 18 is assumed.

1. Sample based policy iteration without bounded rewards

Let the second moments of the rewards given a policy $\pi \in \Pi_S$ exist, i.e.

$$\mathbb{E}_s^\pi[R_0^2] < \infty \ \ \forall s \in \mathcal{S}.$$

Show that the Theorems 4.5.1 and 4.5.2 still apply to this policy, so that the one-step policy evaluation schemes from the lecture converge.

2. Convergence theorem 4.3.8 under weaker assumptions

Show that the statement of Theorem 4.3.8 also holds if $\mathbb{E}[\varepsilon_i(n) \,|\, \mathcal{F}_n] \neq 0$ but instead satisfies

$$\sum_{n=1}^{\infty} \alpha_i(n) \big| \mathbb{E}[\varepsilon_i(n) \,|\, \mathcal{F}_n] \big| < \infty$$

almost surely for all coordinates $i = 1, \dots, d$. It is enough to prove an improved version of Lemma 4.4.4 where the condition $\mathbb{E}[\varepsilon_n \,|\, \mathcal{F}_n] = 0$ is replaced with

$$\sum_{n=1}^{\infty} \alpha_n \big| \mathbb{E}[\varepsilon_n \,|\, \mathcal{F}_n] \big| < \infty. \tag{1}$$

Apply the Robbins-Siegmund theorem to W^2 and use that $W \leq 1 + W^2$.

3. n-step TD

a) Write pseudocode for n-step TD algorithms for evaluation of V^π and Q^π and prove the convergence by checking the n-step Bellman expectation equations

$$T^\pi V(s) = \mathbb{E}_s^\pi \Big[R(s, A_0) + \sum_{t=1}^{n-1} \gamma^t R(S_t, A_t) + \gamma^n V(S_n) \Big]$$

and

$$T^\pi Q(s,a) = \mathbb{E}_s^{\pi^a} \Big[R(s,a) + \sum_{t=1}^{n-1} \gamma^t R(S_t, A_t) + \gamma^n Q(S_n, A_n) \Big]$$

and the conditions of Theorem 4.3.8 on the error term. Note that the algorithm only starts to update after the MDP ran for n steps.

b) Write pseudocode for an n-step SARSA control algorithm.

The solution to the following exercise has to be turned in until 04.05.2026. Groups of up to three people are allowed.

4. *Programming task: Tabular Reinforcement learning

The aim of this exercise is to compare different tabular reinforcement learning algorithms from the lecture. Your task is to run appropriate simulations on adequate environments and present meaningful plots that support your interpretations. Try to compare as many algorithms as possible wherever it is sensible. Include the following:

- Value iteration and Iterative policy evaluation (both for finite-time and discounted MDPs)
- Monte Carlo policy evaluation
- Sample-based policy evaluation
- Q-learning and SARSA,
- General Actor-Critic method

You may choose four out of the six exercises below for you submission, but you are encouraged to do more if you have the time and motivation.

a) Empirically verify the convergence rates of iterative methods that rely on known transition probabilities, comparing them to the convergence speed of sample-based algorithms.

b) Explain the differences between finite-time and discounted MDPs, providing concrete examples to illustrate these distinctions.

c) Define and demonstrate the following effects by constructing „extreme“ MDPs that maximize their visibility, supporting your explanations with appropriate graphs:

- Backpropagation
- Robust Reinforcement Learning
- Overestimation Bias

d) Highlight the significance of selecting appropriate stepsizes and exploration parameters for Q-learning. Develop rules of thumb for choosing schedules based on environment characteristics. Additionally, revisit the concepts of exploration vs. exploitation and commital behavior, interpreting them within the reinforcement learning framework.

e) Compare the general actor-critic method to direct stochastic control algorithms that utilize the Bellman Optimality Operator.

f) Determine the most optimal parameters for Q-learning and Double Q-learning you can find in the following scenario: A 4×4 Grid World with the start state at the top right, the Goal (reward 1) in the bottom row and second column from the left, a Fake Goal (reward 0.65) in the top left, and a stochastic region (reward -2.1/2 with equal probabilities) in the 2×2 square on the bottom right. The default reward is -0.05/0.05 with equal probabilities, and the discount factor is 0.9, both with and without random noise. Document your approach to parameter optimization.

Remember to

- Indicate which Python version and packages you used.
- Label your graphs clearly.
- Specify the (hyper)parameter values chosen for your experiments and algorithms.

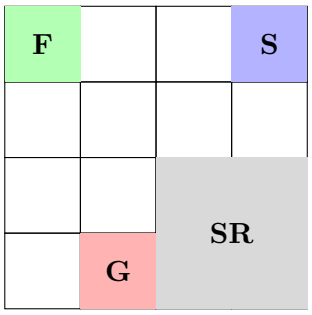

Figure 1: 4×4 Grid World environment. The agent starts at S (top-right). The fake goal F (top-left) is misleading. The actual goal G (bottom row, second from left) gives a reward upon reaching it. The stochastic region (SR) in the bottom right introduces high randomness.

The solution to the theoretical exercises will be discussed in the exercise class in B2 on April 27, 2026, at Mathelounge in B6 B301.